AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) A line cache control system that controls data flow between a line cache, a first central processing unit (CPU) and first and second memory devices, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a first memory interface that communicates with the first memory device;

a second memory interface that communicates with the second memory device;

a line cache that receives a second address that is based on the first address and includes a memory select portion; and

a switch that receives the second address and that selectively connects said line cache to one of said first and second memory interfaces based on the memory select portion,

wherein when said line cache receives said first address, said line cache compares said first address to stored addresses in said line cache, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs.

2. (Currently Amended) The line cache control system of claim 1 wherein said first memory device is ~~RAM~~ random access memory (RAM).

3. (Currently Amended) The line cache control system of claim 2 wherein said RAM is one of ~~DRAM, SDRAM, and DDRAM~~ dynamic RAM (DRAM), synchronous DRAM (SDRAM), and double data rate SDRAM (DDRAM).

4. (Original) The line cache control system of claim 1 wherein said second memory device is flash memory.

5. (Original) The line cache control system of claim 1 wherein said first CPU is an advanced risc machine (ARM) processor.

6. (Original) The line cache control system of claim 1 further comprising:

a second CPU;

a second line cache interface that is associated with said second CPU, that receives a second program read request from said second CPU and that generates a second address from said second program read request; and

a line cache arbitration device that communicates with said first and second line cache interfaces and said line cache and that resolves line cache access conflicts between the first CPU and said second CPU.

7. (Original) The line cache control system of claim 6 further comprising:

a first direct interface that is associated with the first CPU,

wherein said first memory interface includes a second direct interface that communicates with said first direct interface and wherein said first and second direct interfaces allow the first CPU to at least one of read and write data directly to the first memory device.

8. (Original) The line cache control system of claim 7 further comprising:

a third direct interface that is associated with the second CPU,

wherein said second memory interface includes a fourth direct interface that communicates with said third direct interface and wherein said third and fourth direct interfaces allow said second CPU to at least one of read and write data directly to the second memory device.

9. (Original) The line cache control system of claim 8 further comprising a direct read/write arbitration device that resolves direct memory access conflicts between said first and third direct interfaces.

10. (Original) The line cache control system of claim 6 wherein the first CPU is a host processor for a hard disk drive and said second CPU is a servo processor for said hard disk drive.

11. (Original) The line cache control system of claim 1 wherein said line cache includes:

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that includes a line cache state machine that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs.

12. (Currently Amended) A line cache control system that controls data flow between a line cache, a first central processing unit (CPU) and first and second memory devices, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a first memory interface that communicates with the first memory device;

a second memory interface that communicates with the second memory device;

a line cache; and

a switch that selectively connects said line cache to one of said first and second memory interfaces, wherein when said line cache receives said first address, said line cache compares said first address to stored addresses in said line cache, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs,

wherein said line cache includes:

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that includes a line cache state machine that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs; and

~~The line cache control system of claim 11~~ wherein said line cache memory includes multiple pages and wherein said line cache module allows one page to be accessed by one of the first CPU and said second CPU while the other of the first CPU and said second CPU is waiting for data retrieval in another page.

13. (Original) The line cache control system of claim 11 further comprising a least used page device that identifies a least used page in said line cache.

14. (Original) The line cache control system of claim 13 wherein said least used page device replaces said least used page with data retrieved from one of the first and second memory devices when a miss occurs.

15. (Original) The line cache control system of claim 11 wherein state transitions of said line cache state machine are based, in part, on at least one internal state of the CPU.

16. (Currently Amended) A line cache control system that controls data flow between a line cache, a first central processing unit (CPU) and first and second memory devices, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a first memory interface that communicates with the first memory device;

a second memory interface that communicates with the second memory device;

a line cache;

a switch that selectively connects said line cache to one of said first and second memory interfaces, wherein when said line cache receives said first address, said line cache compares said first address to stored addresses in said line cache, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs;

a least used page device that identifies a least used page in said line cache, that replaces said least used page with data retrieved from one of the first and second memory devices when a miss occurs, and that ~~The line cache control system of claim 14 wherein said least used page device~~ identifies a second least used page and wherein said line cache state module checks internal states of the CPU_1

wherein said line cache includes:

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that includes a line cache state machine that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs.

17. (Original) The line cache control system of claim 16 wherein said least used page is replaced when a miss occurs and internal states of the CPU do not indicate a likelihood that said least used page will be needed within a predetermined period.

18. (Original) The line cache control system of claim 17 wherein said second least used page is replaced when a miss occurs and internal states of the CPU indicate a likelihood that said least used page will be needed within a predetermined period.

19. (Currently Amended) A line cache control system that controls data flow between a line cache, a first central processing unit (CPU) and first and second memory devices, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a first memory interface that communicates with the first memory device;

a second memory interface that communicates with the second memory device;

a line cache; and

a switch that selectively connects said line cache to one of said first and second memory interfaces, wherein when said line cache receives said first address, said line cache compares said first address to stored addresses in said line cache, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs,

~~The line cache control system of claim 1~~ wherein said CPU executes an application and wherein said line cache has a line width and number of pages that are based on said application.

20. (Original) The line cache control system of claim 1 wherein said line cache includes 4 pages of 8 x 32.

21. (Currently Amended) A memory control system for a line cache and first and second memory devices that are accessed by a first central processing unit (CPU) and a second CPU, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a second line cache interface that is associated with the second CPU, that receives a second program read request from the second CPU and that generates a second address from said second program read request; ~~and~~

a line cache arbitration device that communicates with said first and second line cache interfaces and said line cache, ~~and~~ that resolves line cache access conflicts between the first CPU and the second CPU, and that generates a translated address based on one of the first address and the second address that includes a memory select portion; and

a switch that receives the translated address and that selectively connects said line cache to one of said first and second memory devices based on the memory select portion.

22. (Currently Amended) The memory control system of Claim 21 wherein said line cache arbitration device selects one of said first and second addresses and further comprising:

a first memory interface that communicates with the first memory device; and

a second memory interface that communicates with the second memory device~~; and~~

~~a switch that selectively connects said line cache to one of said first and second memory interfaces~~,

wherein when said line cache receives said selected one of said first and second addresses from said line cache arbitration device, said line cache compares said selected one of said first and second addresses to stored addresses in said line cache, returns data associated with said selected one of said first and second addresses if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs.

23. (Currently Amended) The memory control system of claim 21 wherein the first memory device is ~~RAM~~ random access memory (RAM) and the second memory device is flash memory.

24. (Previously presented) The memory control system of claim 21 wherein the first CPU is an embedded processor.

25. (Previously presented) The memory control system of claim 21 further comprising:

a first direct interface that is associated with the first CPU, wherein said first memory interface includes a second direct interface that communicates with said first direct interface and wherein said first and second direct interfaces allow the first CPU to at least one of read and write data directly to the first memory device; and

a third direct interface that is associated with the second CPU, wherein said second memory interface includes a fourth direct interface that communicates with said third direct interface and wherein said third and fourth direct interfaces allow the second CPU to at least one of read and write data directly to the second memory device.

26. (Previously presented) The memory control system of claim 25 further comprising a direct read/write arbitration device that resolves direct memory access conflicts between said first and third direct interfaces.

27. (Previously presented) The memory control system of claim 21 wherein the first CPU is a host processor for a hard disk drive and the second CPU is a servo processor for said hard disk drive.

28. (Previously presented) The memory control system of claim 21 wherein said line cache includes:

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs.

29. (Currently Amended) A memory control system for a line cache and first and second memory devices that are accessed by a first central processing unit (CPU) and a second CPU, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a second line cache interface that is associated with the second CPU, that receives a second program read request from the second CPU and that generates a second address from said second program read request;

a line cache arbitration device that communicates with said first and second line cache interfaces and said line cache and that resolves line cache access conflicts between the first CPU and the second CPU;

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that determines when one of a hit and a miss occurs and that manages retrieval of data from the first and second memory devices when said miss occurs, ~~The memory control system of claim 28~~ wherein said line cache memory includes multiple pages and wherein said line cache module allows one page to be accessed by one of the first CPU and the second CPU while the other of the first CPU and the second CPU is waiting for data retrieval in another page.

30. (Previously presented) The memory control system of claim 28 further comprising a least used page device that identifies a least used page in said line cache.

31. (Previously presented) The memory control system of claim 30 wherein said line cache module replaces said least used page with data retrieved from one of the first and second memory devices when said miss occurs.

32. (Currently Amended) A memory control system for a line cache and first and second memory devices that are accessed by a first central processing unit (CPU) and a second CPU, comprising:

a first line cache interface that is associated with the first CPU, that receives a first program read request from the first CPU and that generates a first address from said first program read request;

a second line cache interface that is associated with the second CPU, that receives a second program read request from the second CPU and that generates a second address from said second program read request;

a line cache arbitration device that communicates with said first and second line cache interfaces and said line cache and that resolves line cache access conflicts between the first CPU and the second CPU;

line cache memory that stores data;

a content addressable memory (CAM) that stores addresses associated with said data stored in said line cache memory; and

a line cache module that determines when one of a hit and a miss occurs, that manages retrieval of data from the first and second memory devices when said miss occurs, and that replaces said least used page with data retrieved from one of the first and second memory devices when said miss occurs; and

a least used page device that identifies a least used page in said line cache, ~~The memory control system of claim 31~~ wherein said line cache module monitors at least one internal state of the first CPU, wherein said least used page device identifies a second least used page, and wherein said line cache state module

selectively overwrites one of said first and second least used pages said based on internal state of the first CPU.

33. (Previously presented) The memory control system of claim 32 wherein said least used page is overwritten when said miss occurs and said line cache module determines that said least used page will not be requested by the first CPU within a predetermined period based one said at least on internal state of the first CPU.

34. (Previously presented) The memory control system of claim 32 wherein said second least used page is overwritten when said miss occurs and said line cache module determines that said least used page will be requested by the first CPU within a predetermined period based on said at least one internal state of the first CPU.

35. (Previously presented) A line cache system that communicates with a first central processing unit (CPU), and a first memory device comprising:

line cache memory that stores data as pages;

a content addressable memory (CAM) that stores addresses associated with said data stored in said pages of said line cache memory; and

a line cache module that monitors at least one internal state of the first CPU, that determines when one of a hit and a miss occurs and that manages retrieval of data from the first memory device when said miss occurs,

wherein said line cache module includes a least used page device that selects first and second pages to be overwritten and wherein said line cache module selects one of said first and second pages based on said internal state of the first CPU.

36. (Previously presented) The line cache control system of Claim 35 wherein said first and second pages identified by said least used page device are first and second least used pages in said line cache.

37. (Previously presented) The line cache control system of Claim 36 wherein when said miss occurs and said line cache module determines that said first least used page will be used by the first CPU within a predetermined period based on said at least one internal state of the first CPU, said line cache module overwrites said second least used page using data retrieved from the first memory device.

38. (Previously presented) The line cache control system of Claim 36 wherein when said miss occurs and said line cache module determines that said first least used page will not be used by the first CPU within a predetermined period based on said at least one internal state of the first CPU, said line cache module overwrites said first least used page using data retrieved from the first memory device.

39. (Previously presented) A line cache system that communicates with a first central processing unit (CPU) and a second CPU and a first memory device, comprising:

line cache memory that stores data in a plurality of pages;

a content addressable memory (CAM) that stores addresses associated with said data stored in said pages of said line cache memory; and

a line cache module that determines when one of a hit and a miss occurs and that manages retrieval of data from the first memory device when said miss occurs, wherein said line cache module allows one of said pages to be accessed by one of the first CPU and the second CPU while the other of the first CPU and the second CPU is waiting for data retrieval to another of said pages.

40. (Previously presented) The line cache control system of claim 39 wherein said line cache module monitors at least one internal state of the first CPU and selects at least one of said pages for replacement based on usage of said pages and said at least one internal state of said CPU.

41. (Previously presented) The line cache control system of Claim 40 wherein said line cache module includes a least used page device that identifies first and second least used pages in said line cache.

42. (Previously presented) The line cache control system of Claim 41 wherein when said miss occurs and said line cache module determines that said first least used page will be used by the first CPU within a predetermined period based on said at least one internal state of the first CPU, said line cache module overwrites said second least used page using data retrieved from the first memory device.

43. (Previously presented) The line cache control system of Claim 41 wherein when said miss occurs and said line cache module determines that said first least used page will not be used by the first CPU within a predetermined period based on said at least one internal state of the first CPU, said line cache module overwrites said first least used page using data retrieved from the first memory device.

44. (Currently Amended) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

generating a translated address that includes a memory select portion based on said first address;

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively, based on said memory select portion;

comparing said first address to stored addresses in said line cache when said line cache receives said first address;

returning data associated with said first address if a match occurs; and

retrieving data from one of said first and second memory devices if a miss occurs.

45. (Currently Amended) The method of claim 44 wherein said first memory device is ~~RAM~~ random access memory (RAM).

46. (Currently Amended) The method of claim 45 wherein said RAM is one of ~~DRAM, SDRAM, and DDRAM~~ dynamic RAM (DRAM), synchronous DRAM (SDRAM), and double data rate SDRAM (DDRAM).

47. (Previously presented) The method of claim 44 wherein said second memory device is flash memory.

48. (Previously presented) The method of claim 44 wherein said first CPU is an embedded processor.

49. (Previously presented) The method of claim 44 further comprising:

receiving a second program read request from a second CPU at a second line cache interface;

generating a second address from said second program read request; and

resolving line cache access conflicts between said first and second CPUs.

50. (Previously presented) The method of claim 44 further comprising:

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs; and

managing retrieval of data from said first and second memory devices when said miss occurs.

51. (Currently Amended) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively;

comparing said first address to stored addresses in said line cache when said line cache receives said first address;

returning data associated with said first address if a match occurs;

retrieving data from one of said first and second memory devices if a miss occurs;

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs; and

managing retrieval of data from said first and second memory devices when said miss occurs,

~~The method of claim 50~~ wherein said line cache memory includes multiple pages and further comprising allowing one page to be accessed by one of said first CPU and said second CPU while the other of said first CPU and said second CPU is waiting for data retrieval in another page.

52. (Previously presented) The method of claim 50 further comprising identifying a first least used page in said line cache.

53. (Previously presented) The method of claim 52 further comprising replacing said first least used page with data retrieved from one of the first and second memory devices when a miss occurs.

54. (Previously presented) The method of claim 50 further comprising operating said line cache based on at least one internal state of said first CPU.

55. (Currently Amended) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively;

comparing said first address to stored addresses in said line cache when said line cache receives said first address;

returning data associated with said first address if a match occurs;

retrieving data from one of said first and second memory devices if a miss occurs;

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs;

managing retrieval of data from said first and second memory devices when said miss occurs;

identifying a first least used page in said line cache;

~~The method of claim 52 further comprising:~~

identifying a second least used page; and

checking at least one internal state of said first CPU.

56. (Previously presented) The method of claim 55 further comprising replacing said first least used page when a miss occurs and said at least one internal state of said first CPU do not indicate a likelihood that said first least used page will be needed within a predetermined period.

57. (Previously presented) The method of claim 55 further comprising replacing said second least used page when a miss occurs and said at least one internal state of said first CPU indicate a likelihood that said first least used page will be needed within a predetermined period.

58. (Currently Amended) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively;

comparing said first address to stored addresses in said line cache when said line cache receives said first address;

returning data associated with said first address if a match occurs; and

retrieving data from one of said first and second memory devices if a miss occurs,

~~The method of claim 44~~ wherein said first CPU executes an application and further comprising basing a line width and number of pages of said line cache on said application.

59. (Currently Amended) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

receiving a second program read request from a second CPU at a second line cache interface;

generating a second address from said second program read request; ~~and~~

generating a translated address based on at least one of the first address and the second address, wherein the translated address includes a memory select portion;

resolving line cache access conflicts between said first CPU and said second CPU; and

selectively connecting said line cache to one of first and second memory interfaces for first and second memory devices, respectively, based on the memory select portion.

60. (Currently Amended) The method of Claim 59 further comprising:

selecting one of said first and second addresses;

~~selectively connecting said line cache to one of a first memory interface for a first memory device and a second memory interface for a second memory device;~~

comparing said selected one of said first and second addresses to stored addresses in said line cache;

returning data associated with said selected one of said first and second addresses if a match occurs; and

retrieving data from one of said first and second memory devices if a miss occurs.

61. (Currently Amended) The method of Claim 59 wherein said first memory device is ~~RAM~~ random access memory (RAM) and said second memory device is flash memory.

62. (Previously presented) The method of Claim 59 wherein said first CPU is an embedded processor.

63. (Previously presented) The method of Claim 59 further comprising:

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs; and

managing retrieval of data from said first and second memory devices when said miss occurs.

64. (Currently Amended) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

receiving a second program read request from a second CPU at a second line cache interface;

generating a second address from said second program read request;

resolving line cache access conflicts between said first CPU and said second CPU;

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs; and

managing retrieval of data from said first and second memory devices when said miss occurs,

~~The method of Claim 63~~ wherein said line cache memory includes multiple pages and further comprising allowing one page to be accessed by one of said first CPU and said second CPU while the other of said first CPU and said second CPU is waiting for data retrieval in another page.

65. (Previously presented) The method of Claim 63 further comprising identifying a first least used page in said line cache.

66. (Previously presented) The method of Claim 65 further comprising replacing said first least used page with data retrieved from one of the first and second memory devices when said miss occurs.

67. (Currently Amended) A method for operating a line cache, comprising:

receiving a first program read request from a first CPU at a first line cache interface;

generating a first address from said first program read request;

receiving a second program read request from a second CPU at a second line cache interface;

generating a second address from said second program read request;

resolving line cache access conflicts between said first CPU and said second CPU;

storing data in line cache memory;

storing addresses associated with said data stored in said line cache memory;

determining when one of a hit and a miss occurs;

managing retrieval of data from said first and second memory devices when said miss occurs;

identifying a first least used page in said line cache;

~~The method of Claim 65 further comprising:~~

monitoring at least one internal state of said first CPU;

identifying a second least used page; and

selectively overwriting one of said first and second least used pages based on said at least one internal state of said first CPU.

68. (Previously presented) The method of Claim 67 further comprising overwriting said first least used page when said miss occurs and when said first least used page will not be requested by said first CPU within a predetermined period based one said at least one internal state of said first CPU.

69. (Previously presented) The method of Claim 67 further comprising overwriting said first least used page when said miss occurs and when said first least used page will be requested by said first CPU within a predetermined period based on said at least one internal state of said first CPU.

70. (Previously presented) A method for operating a line cache system that communicates with a first central processing unit (CPU) and a first memory device, comprising:

storing data as pages in line cache memory;

storing addresses associated with said data stored in said pages of said line cache memory;

monitoring at least one internal state of said first CPU;

determining when one of a hit and a miss occurs;

managing retrieval of data from said first memory device when said miss occurs; and

selecting one of said first and second pages to be overwritten based on said internal state of said first CPU.

71. (Previously presented) The method of Claim 70 wherein said first and second pages are first and second least used pages in said line cache.

72. (Previously presented) The method of Claim 71 further comprising overwriting said second least used page using data retrieved from said first memory device when said miss occurs and said first least used page will be used by said first CPU within a predetermined period based on said at least one internal state of said first CPU.

73. (Previously presented) The method of Claim 71 further comprising overwriting said first least used page using data retrieved from said first memory device when said miss occurs and said first least used page will not be used by said first CPU within a predetermined period based on said at least one internal state of said first CPU.

74. (Previously presented) A method for operating a line cache system that communicates with a first central processing unit (CPU) and a second CPU and a first memory device, comprising:

storing data in a plurality of pages of line cache memory;

storing addresses associated with said data stored in said pages of said line cache memory;

determining when one of a hit and a miss occurs;

managing retrieval of data from said first memory device when said miss occurs; and

allowing one of said pages to be accessed by one of said first CPU and said second CPU while the other of said first CPU and said second CPU is waiting for data retrieval to another of said pages.

75. (Previously presented) The method of claim 74 further comprising:

monitoring at least one internal state of said first CPU; and

selecting at least one of said pages for replacement based on usage of said pages and said at least one internal state of said first CPU.

76. (Previously presented) The method of Claim 74 further comprising identifying first and second least used pages in said line cache.

77. (Previously presented) The method of Claim 76 further comprising overwriting said second least used page using data retrieved from said first memory device when said miss occurs and said first least used page will be used by said first CPU within a predetermined period based on said at least one internal state of said first CPU.

78. (Previously presented) The method of Claim 76 further comprising overwriting said first least used page using data retrieved from said first memory device when said miss occurs and said first least used page will not be used by said first CPU within a predetermined period based on said at least one internal state of said first CPU.

79. (Currently Amended) A line cache control system comprising:

first processing means for processing data;

first and second memory means for storing data;

first line cache interface means that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request;

first memory interface means for communicating with said first memory means;

second memory interface means for communicating with said second memory means;

line cache means for storing data and receiving a translated address based on the first address that includes a memory select portion; and

selecting means for receiving the translated address and selectively connecting said line cache means to one of said first and second memory interface means based on the memory select portion,

wherein when said line cache means receives said first address, said line cache means compares said first address to stored addresses in said line cache means, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs.

80. (Currently Amended) The line cache control system of claim 79 wherein said first memory means is ~~RAM~~ random access memory (RAM).

81. (Currently Amended) The line cache control system of claim 80 wherein said RAM is one of ~~DRAM, SDRAM, and DDRAM~~ dynamic RAM (DRAM), synchronous DRAM (SDRAM), and double data rate SDRAM (DDRAM).

82. (Previously presented) The line cache control system of claim 79 wherein said second memory means is flash memory.

83. (Previously presented) The line cache control system of claim 79 wherein said first processing means is an embedded processor.

84. (Previously presented) The line cache control system of claim 79 further comprising:

second processing means for processing data;

second line cache interface means that is associated with said second processing means, that receives a second program read request from said second processing means and that generates a second address from said second program read request; and

line cache arbitration means that communicates with said first and second line cache interface means for resolving access conflicts to said line cache means between said first processing means and said second processing means.

85. (Previously presented) The line cache control system of claim 84 further comprising:

first direct interface means that is associated with said first processing means,

wherein said first memory interface means includes a second direct interface means that communicates with said first direct interface means and wherein said first and second direct interface means allow said first processing means to at least one of read and write data directly to said first memory means.

86. (Previously presented) The line cache control system of claim 85 further comprising:

third direct interface means that is associated with said second processing means,

wherein said second memory interface means includes a fourth direct interface means that communicates with said third direct interface means and wherein said third and fourth direct interface means allow said second processing means to at least one of read and write data directly to said second memory means.

87. (Previously presented) The line cache control system of claim 86 further comprising direct read/write arbitration means for resolving direct memory access conflicts between said first and third direct interface means.

88. (Previously presented) The line cache control system of claim 84 wherein said first processing means is a host processor for a hard disk drive and said second processing means is a servo processor for said hard disk drive.

89. (Previously presented) The line cache control system of claim 79 wherein said line cache means includes:

data storing means for storing data;

content addressable memory means for storing addresses associated with said data stored in said data storing means; and

line cache control means that determines when one of a hit and a miss occurs and that manages retrieval of data from said first and second memory means when said miss occurs.

90. (Currently Amended) A line cache control system comprising:

first processing means for processing data;

first and second memory means for storing data;

first line cache interface means that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request;

first memory interface means for communicating with said first memory means;

second memory interface means for communicating with said second memory means;

line cache means for storing data that includes:

data storing means for storing data;

content addressable memory means for storing addresses associated with said data stored in said data storing means; and

line cache control means that determines when one of a hit and a miss occurs and that manages retrieval of data from said first and second memory means when said miss occurs; and

selecting means for selectively connecting said line cache means to one of said first and second memory interface means,

wherein when said line cache means receives said first address, said line cache means compares said first address to stored addresses in said line cache means, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs; and

~~The line cache control system of claim 89~~ wherein said line cache means includes multiple pages and wherein said line cache control means allows one page to be accessed by one of said first and second processing means while the other of said first and second processing means is waiting for data retrieval in another page.

91. (Previously presented) The line cache control system of claim 89 further comprising least used page means for identifying a first least used page in said line cache means.

92. (Previously presented) The line cache control system of claim 91 wherein said line cache control means replaces said first least used page with data retrieved from one of the first and second memory means when a miss occurs.

93. (Previously presented) The line cache control system of claim 92 wherein said least used page means identifies a second least used page and wherein said line cache control means checks internal states of said first processing means.

94. (Currently Amended) A line cache control system comprising:

first processing means for processing data;

first and second memory means for storing data;

first line cache interface means that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request;

first memory interface means for communicating with said first memory means;

second memory interface means for communicating with said second memory means;

line cache means for storing data that includes:

data storing means for storing data;

content addressable memory means for storing addresses associated with said data stored in said data storing means; and

line cache control means that determines when one of a hit and a miss occurs and that manages retrieval of data from said first and second memory means when said miss occurs;

selecting means for selectively connecting said line cache means to one of said first and second memory interface means;

least used page means for identifying a first least used page in said line cache means

wherein when said line cache means receives said first address, said line cache means compares said first address to stored addresses in said line cache

means, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs, said line cache control means replaces said first least used page with data retrieved from one of the first and second memory means when a miss occurs, said least used page means identifies a second least used page and wherein said line cache control means checks internal states of said first processing means, and ~~The line cache control system of claim 93 wherein~~ said first least used page is replaced when a miss occurs and internal states of said first processing means do not indicate a likelihood that said first least used page will be needed within a predetermined period.

95. (Previously presented) The line cache control system of claim 94 wherein said second least used page is replaced when a miss occurs and internal states of said first processing means indicate a likelihood that said first least used page will be needed within a predetermined period.

96. (Currently Amended) A line cache control system comprising:

first processing means for processing data;

first and second memory means for storing data;

first line cache interface means that is associated with said first processing means for receiving a first program read request from said first processing means and for generating a first address from said first program read request;

first memory interface means for communicating with said first memory means;

second memory interface means for communicating with said second memory means;

line cache means for storing data; and

selecting means for selectively connecting said line cache means to one of said first and second memory interface means,

wherein when said line cache means receives said first address, said line cache means compares said first address to stored addresses in said line cache means, returns data associated with said first address if a match occurs, and retrieves data from one of the first and second memory means if a miss occurs, and ~~The line cache control system of claim 79 wherein~~ said first processing means executes an application and wherein said line cache has a line width and number of pages that are based on said application.

97. (Previously presented) The line cache control system of claim 79 wherein said line cache includes 4 pages of 8 x 32.

98. (Currently Amended) A memory control system for a line cache and first and second memory devices that are accessed by a first central processing unit (CPU) and a second CPU, comprising:

first line cache interface means that is associated with the first CPU, for receiving a first program read request from the first CPU and for generating a first address from said first program read request;

second line cache interface means that is associated with the second CPU, for receiving a second program read request from the second CPU and for generating a second address from said second program read request; ~~and~~

line cache arbitration means that communicates with said first and second line cache interface means and said line cache for resolving line cache access conflicts between the first CPU and the second CPU and for generating a translated address based on one of the first address and the second address that includes a memory select portion; and

switching means for receiving the translated address and for selectively connecting said line cache to one of said first and second memory devices based on the memory select portion.

99. (Currently Amended) The memory control system of Claim 98 wherein said line cache arbitration means selects one of said first and second addresses and further comprising:

first memory interface means for communicating with the first memory device; and

second memory interface means for communicating with the second memory device~~; and~~

~~selecting means for selectively connecting said line cache to one of said first and second memory interface means~~,

wherein when said line cache receives said selected one of said first and second addresses from said line cache arbitration means, said line cache compares said selected one of said first and second addresses to stored addresses in said line cache, returns data associated with said selected one of said first and second addresses if a match occurs, and retrieves data from one of the first and second memory devices if a miss occurs.

100. (Currently Amended) The memory control system of claim 98 wherein the first memory device is ~~RAM~~ random access memory (RAM) and the second memory device is flash memory.

101. (Previously presented) The memory control system of claim 98 wherein the first CPU is an embedded processor.

102. (Previously presented) The memory control system of claim 98 further comprising:

first direct interface means that is associated with the first CPU, wherein said first memory interface means includes a second direct interface means for communicating with said first direct interface means and wherein said first and second direct interface means allow the first CPU to at least one of read and write data directly to the first memory device; and

a third direct interface means that is associated with the second CPU, wherein said second memory interface means includes a fourth direct interface means for communicating with said third direct interface means and wherein said third and fourth direct interface means allow the second CPU to at least one of read and write data directly to the second memory device.

103. (Previously presented) The memory control system of claim 102 further comprising direct read/write arbitration means for resolving direct memory access conflicts between said first and third direct interface means.

104. (Previously presented) The memory control system of claim 98 wherein the first CPU is a host processor for a hard disk drive and the second CPU is a servo processor for said hard disk drive.

105. (Previously presented) The memory control system of claim 98 wherein said line cache includes:

data storing means for storing data;

content addressable memory (CAM) means for storing addresses associated with said data stored in said storing means; and

line cache control means for determining when one of a hit and a miss occurs and for managing retrieval of data from the first and second memory devices when said miss occurs.

106. (Currently Amended) A memory control system for a line cache and first and second memory devices that are accessed by a first central processing unit (CPU) and a second CPU, comprising:

first line cache interface means that is associated with the first CPU, for receiving a first program read request from the first CPU and for generating a first address from said first program read request;

second line cache interface means that is associated with the second CPU, for receiving a second program read request from the second CPU and for generating a second address from said second program read request; and

line cache arbitration means that communicates with said first and second line cache interface means and said line cache for resolving line cache access conflicts between the first CPU and the second CPU,

wherein said line cache includes:

data storing means for storing data;

content addressable memory (CAM) means for storing addresses associated with said data stored in said storing means; and

line cache control means for determining when one of a hit and a miss occurs and for managing retrieval of data from the first and second memory devices when said miss occurs,

and ~~The memory control system of claim 105~~ wherein said data storing means includes multiple pages and wherein said line cache control means allows one page to be accessed by one of the first and second CPUs while the other of the first and second CPUs is waiting for data retrieval in another page.

107. (Previously presented) The memory control system of claim 105 further comprising least used page means for identifying a first least used page in said line cache.

108. (Previously presented) The memory control system of claim 107 wherein said line cache control means replaces said first least used page with data retrieved from one of the first and second memory devices when said miss occurs.

109. (Currently Amended) A memory control system for a line cache and first and second memory devices that are accessed by a first central processing unit (CPU) and a second CPU, comprising:

first line cache interface means that is associated with the first CPU, for receiving a first program read request from the first CPU and for generating a first address from said first program read request;

second line cache interface means that is associated with the second CPU, for receiving a second program read request from the second CPU and for generating a second address from said second program read request; and

line cache arbitration means that communicates with said first and second line cache interface means and said line cache for resolving line cache access conflicts between the first CPU and the second CPU; and

least used page means for identifying a first least used page in said line cache,

wherein said line cache includes:

data storing means for storing data;

content addressable memory (CAM) means for storing addresses associated with said data stored in said storing means; and

line cache control means for determining when one of a hit and a miss occurs and for managing retrieval of data from the first and second memory devices when said miss occurs, and

wherein said line cache control means replaces said first least used page with data retrieved from one of the first and second memory devices when said

miss occurs, ~~The memory control system of claim 108 wherein~~ said line cache control means monitors at least one internal state of the first CPU, ~~wherein~~ said least used page means identifies a second least used page, and ~~wherein~~ said line cache control means selectively overwrites one of said first and second least used pages based on said at least one internal state of the first CPU.

110. (Previously presented) The memory control system of claim 109 wherein said first least used page is overwritten when said miss occurs and said line cache control means determines that said least used page will not be requested by the first CPU within a predetermined period based one said at least one internal state of the first CPU.

111. (Previously presented) The memory control system of claim 109 wherein said second least used page is overwritten when said miss occurs and said line cache control means determines that said least used page will be requested by the first CPU within a predetermined period based on said at least one internal state of the first CPU.

112. (Previously presented) A line cache system that communicates with a first central processing unit (CPU) and a first memory device, comprising:

line cache memory means for storing data as pages;

content addressable memory (CAM) means for storing addresses associated with said data stored in said pages of said line cache memory means; and

line cache control means for monitoring at least one internal state of the first CPU, for determining when one of a hit and a miss occurs and for managing retrieval of data from the first memory device when said miss occurs,

wherein said line cache control means includes page selecting means for selecting first and second pages to be overwritten and wherein said line cache control means selects one of said first and second pages based on said at least one internal state of the first CPU.

113. (Previously presented) The line cache control system of Claim 112 wherein said first and second pages identified by said page selecting means are first and second least used pages in said line cache.

114. (Previously presented) The line cache control system of Claim 113 wherein when said miss occurs and said line cache control means determines that said first least used page will be used by the first CPU within a predetermined period based on said at least one internal state of the first CPU, said line cache control means overwrites said second least used page using data retrieved from the first memory device.

115. (Previously presented) The line cache control system of Claim 113 wherein when said miss occurs and said line cache control means determines that said first least used page will not be used by the first CPU within a predetermined period based on said at least one internal state of the first CPU, said line cache control means overwrites said first least used page using data retrieved from the first memory device.

116. (Previously presented) A line cache system that communicates with a first central processing unit (CPU) and a second CPU and a first memory device, comprising:

line cache memory means for storing data in a plurality of pages;

content addressable memory (CAM) means for storing addresses associated with said data stored in said pages of said line cache memory means; and

line cache control means for determining when one of a hit and a miss occurs and for managing retrieval of data from the first memory device when said miss occurs, wherein said line cache control means allows one of said pages to be accessed by one of the first and second CPUs while the other of the first and second CPUs is waiting for data retrieval to another of said pages.

117. (Previously presented) The line cache control system of claim 116 wherein said line cache control means monitors at least one internal state of the first CPU and selects at least one of said pages for replacement based on usage of said pages and said at least one internal state of said CPU.

118. (Previously presented) The line cache control system of Claim 117 wherein said line cache control means includes least used page means that identifies first and second least used pages in said line cache.

119. (Previously presented) The line cache control system of Claim 118 wherein when said miss occurs and said line cache control means determines that said first least used page will be used by the first CPU within a predetermined period based on said at least one internal state of the first CPU, said line cache control means overwrites said second least used page using data retrieved from the first memory device.

120. (Previously presented) The line cache control system of Claim 118 wherein when said miss occurs and said line cache control means determines that said first least used page will not be used by the first CPU within a predetermined period based on said at least one internal state of the first CPU, said line cache control means overwrites said first least used page using data retrieved from the first memory device.